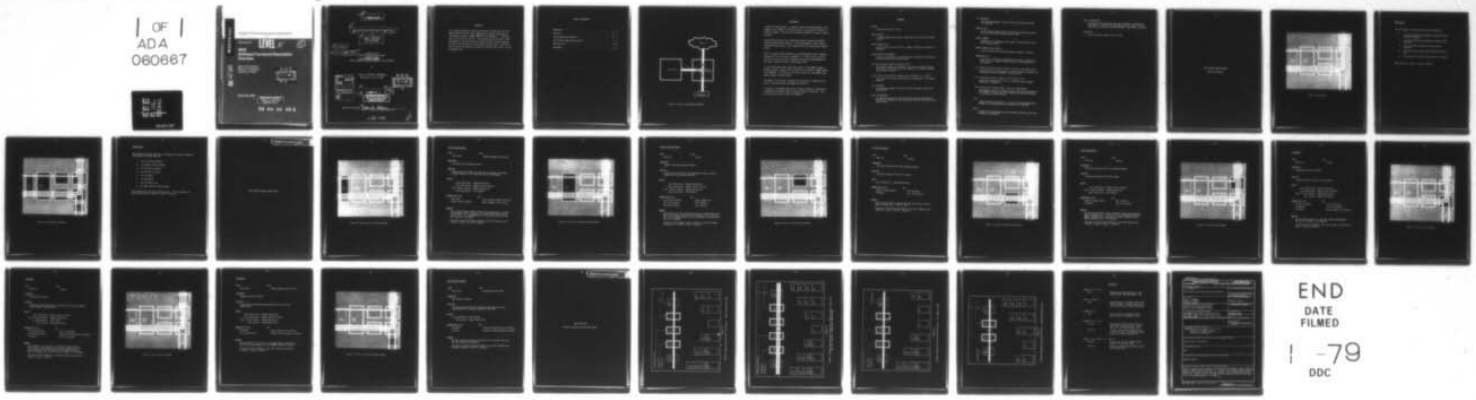
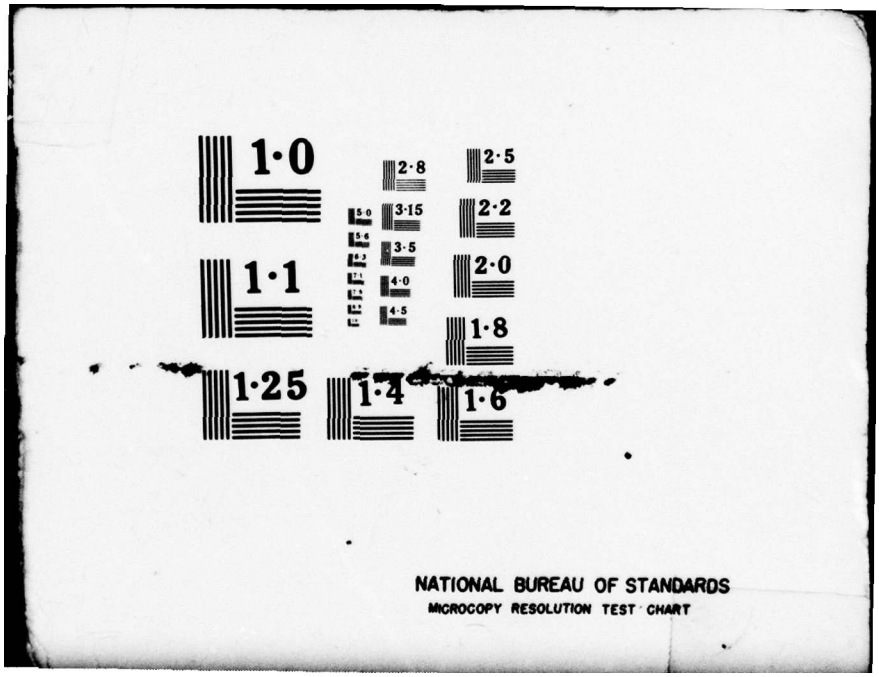
AD-A060 667 DIGITAL TECHNOLOGY INC CHAMPAIGN IL F/G 9/2
INFE SOFTWARE FUNCTIONAL DESCRIPTION OVERVIEW,(U)
MAR 78 G R GROSSMAN, S F HOLMGREN, R H HOWE DCA100-77-C-0069
UNCLASSIFIED DTI-2 SBIE-AD-E100 081 NL



END
DATE FILMED
1-79
DDC


1·0
1·1
1·25
2·8
3·15
3·5
4·0
4·5
5·0
2·5
2·2
2·0
1·8
1·4
1·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

Software Functional Description Overview

Digital Technology Incorporated

Document 2

LEVEL II

(1)

INFE
Software Functional Description
Overview

Gary R. Grossman
Steven F. Holmgren
Richard H. Howe

March 20, 1978

(14) DTI-[redacted]-2

(6) # INFE SOFTWARE FUNCTIONAL DESCRIPTION OVERVIEW

by

(10) Gary R. Grossman
Steven F. Holmgren
Richard H. Howe

(18) SBIE

(19) AD-E100 081

Prepared for the
Defense Communications Agency
under contract
(15) DCA100-77-C-0069
Phase B Network Front-End Research and Development

by

DIGITAL TECHNOLOGY INCORPORATED
Champaign, Illinois 61820

(11) 29 Mar[redacted] 78

(12) 38 p.

Approved for Release: Peter A. Alsberg
Peter A. Alsberg, President

410 796

ABSTRACT

This document presents a functional description overview of the IOC Network Frontend (INFE) under development by Digital Technology Incorporated to connect a WWMCCS H6000 host to AUTODIN II. A glossary and a brief history are given. The architecture of the INFE is then presented as a set of data routes through the INFE protocol modules. The INFE protocol modules are described in tabular form. The flow of data vis-a-vis the staging of protocols along the INFE routes is given diagrammatically.

TABLE OF CONTENTS

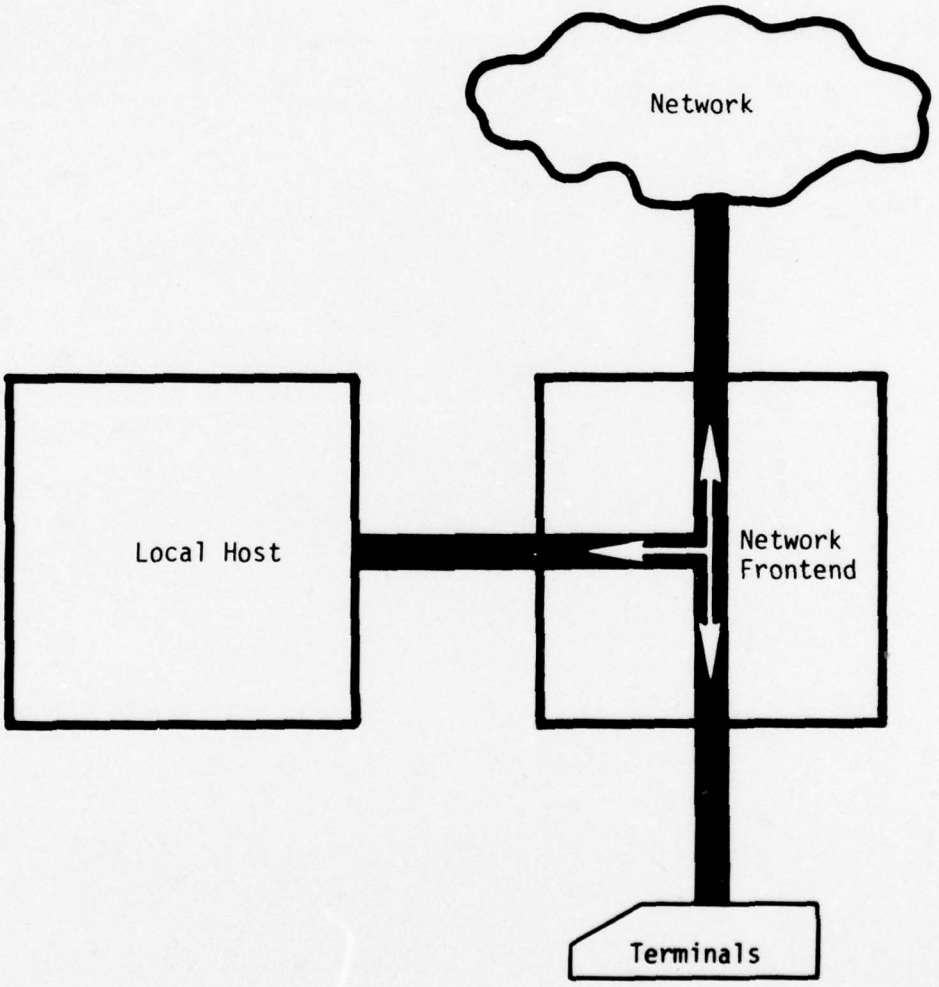


Figure 1: Place of the Network Frontend

BACKGROUND

A Network Frontend (NFE) is a computer system interposed between a host computer and a network to relieve the host of the burden of network interface software. The place of the Network Frontend is shown in Figure 1, opposite.

The technological basis for a Network Frontend is derived from ARPA network access machine research. A late development in that research was Network UNIX, which is the immediate basis for the present research effort.

UNIX is a general purpose operating system developed by Bell Telephone Laboratories for use on PDP-11 computers. In 1975, staff of the University of Illinois Center for Advanced Computation added ARPANET NCP and Telnet software to UNIX to make Network UNIX.

In 1976, the Network UNIX staff began Phase A of the WWMCCS Network Frontend project. Network UNIX was modified to frontend a WWMCCS H6000 to the ARPANET. The result of the Phase A effort was the WWMCCS Experimental Network Frontend (ENFE).

In October, 1977, Digital Technology Incorporated was formed from the staff of the Network UNIX and WWMCCS NFE projects.

In Phase B of the WWMCCS NFE program, Digital Technology Incorporated is modifying the Phase A ENFE to support AUTODIN II protocols. This will provide an IOC NFE (INFE) for AUTODIN II.

GLOSSARY

DEVICE:

a named data source or sink.

DEVICE DRIVER:

a software module that enables a UNIX user-level process to access a device.

DEVICE DRIVER TYPE:

a set of device drivers sharing a common interface to the UNIX I/O subsystem.

DEVICE DRIVER TYPE HANDLER:

the layer of the UNIX I/O subsystem which implements the interfaces to the different device driver types.

HFP: Host-Frontend Protocol (Grossman 1976):

HFP defines communication between a host and a frontend. HFP has three levels: the link level, the channel level, and the process-service level protocols.

IIPC: Illinois Inter-Process Communication (Holmgren et al. 1977b):

IIPC enables UNIX user-level processes to communicate using memory segments.

IIPC INTERFACE:

the interface between two UNIX user-level processes directly employing IIPC.

IIPC I/O INTERFACE:

an interface between two UNIX user-level processes which employs the UNIX I/O interface at one process and the IIPC interface at the other process.

I/O INTERFACE:

the interface between a UNIX user-level process and the UNIX I/O Subsystem.

KERNEL LEVEL:

the privileged software level of UNIX which includes the device drivers and the UNIX operating system itself.

MEMORY SEGMENT:

a named set of contiguous virtual memory locations which can be shared among processes.

MEMORY SEGMENT DEVICE DRIVER:

a type of device driver employing memory segments as buffers.

NONBLOCKING I/O:

a facility of the UNIX I/O subsystem which enables a UNIX user-level process to proceed without waiting for the completion of I/O requests.

SIP: Segment Interface Protocol (Postel, Garlick, & Rom 1976a:36-39):

the protocol defining the rules for transmitting data between TCP implementations and an AUTODIN II packet switch.

TCP: Transmission Control Protocol (Cerf & Postel 1977):

the protocol defining the rules for transmitting data between processes via AUTODIN II.

THP: Terminal-Host Protocol (Postel, Garlick, & Rom 1976b):

the protocol defining the network virtual terminal representation for AUTODIN II and the rules for communication between instances of the network virtual terminal.

VIP:

"Video Information Projection". A class of CRT terminals made by Honeywell; the standard data entry terminal for the H6000.

UNIX:

the operating system comprising the substrate upon which the INFE software is constructed.

UNIX I/O SUBSYSTEM:

the mechanism enabling UNIX user-level processes to communicate with devices. The UNIX I/O Subsystem has three layers: the I/O interface, the device driver type handlers, and the device drivers.

USER LEVEL:

the unprivileged software level of UNIX.

INFE SOFTWARE ARCHITECTURE:

Routes and Modules

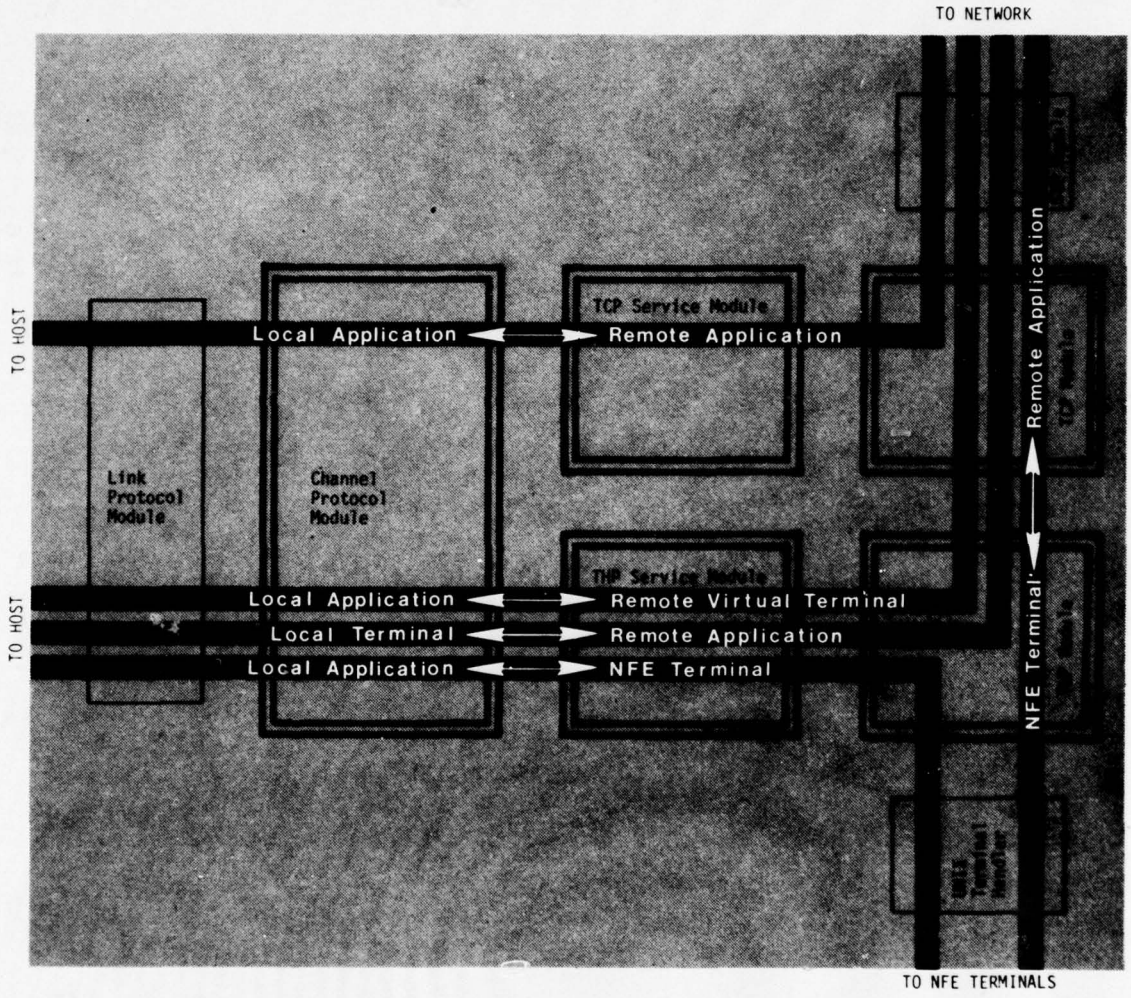


Figure 2: INFE Routes

INFE Routes

The INFE manages data flowing along routes connecting:

1. local host application programs and remote host application programs,
2. local host application programs and remote virtual terminals,
3. local terminals and remote host application programs,
4. local host application programs and NFE terminals, and
5. NFE terminals and remote host application programs.

These routes are shown in Figure 2, opposite.

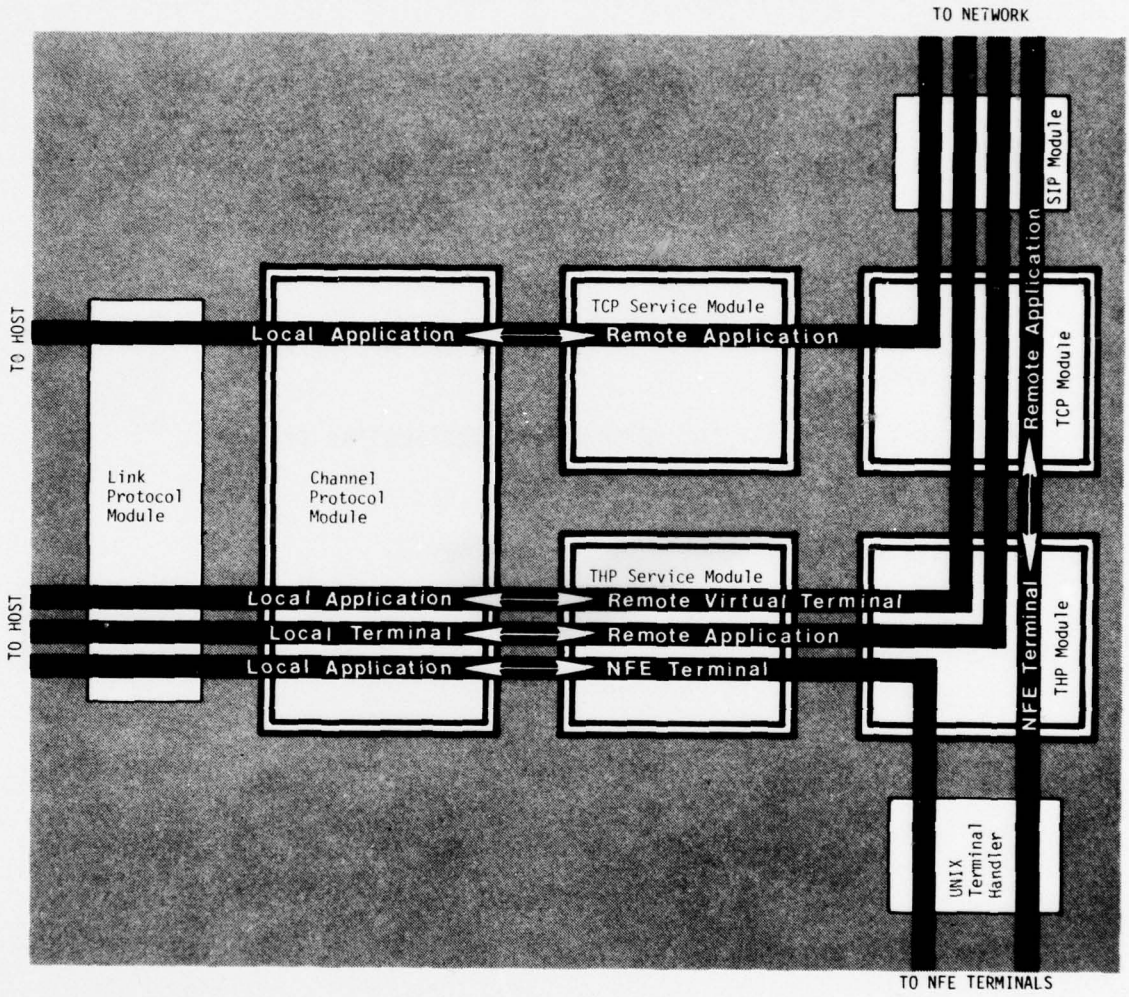


Figure 3: INFE Routes and Modules

INFE Modules

Data flowing along the INFE routes is managed via stages of protocol implemented by software modules:

1. the Link Protocol Module,
2. the Channel Protocol Module,
3. the TCP Service Module,
4. the THP Service Module,
5. the TCP Module,
6. the THP Module,
7. the SIP Module, and
8. the UNIX Terminal Handler Module.

These modules may serve more than one route. The relationships of routes and modules are shown in Figure 3, opposite.

INFE PROTOCOL MODULE DESCRIPTIONS

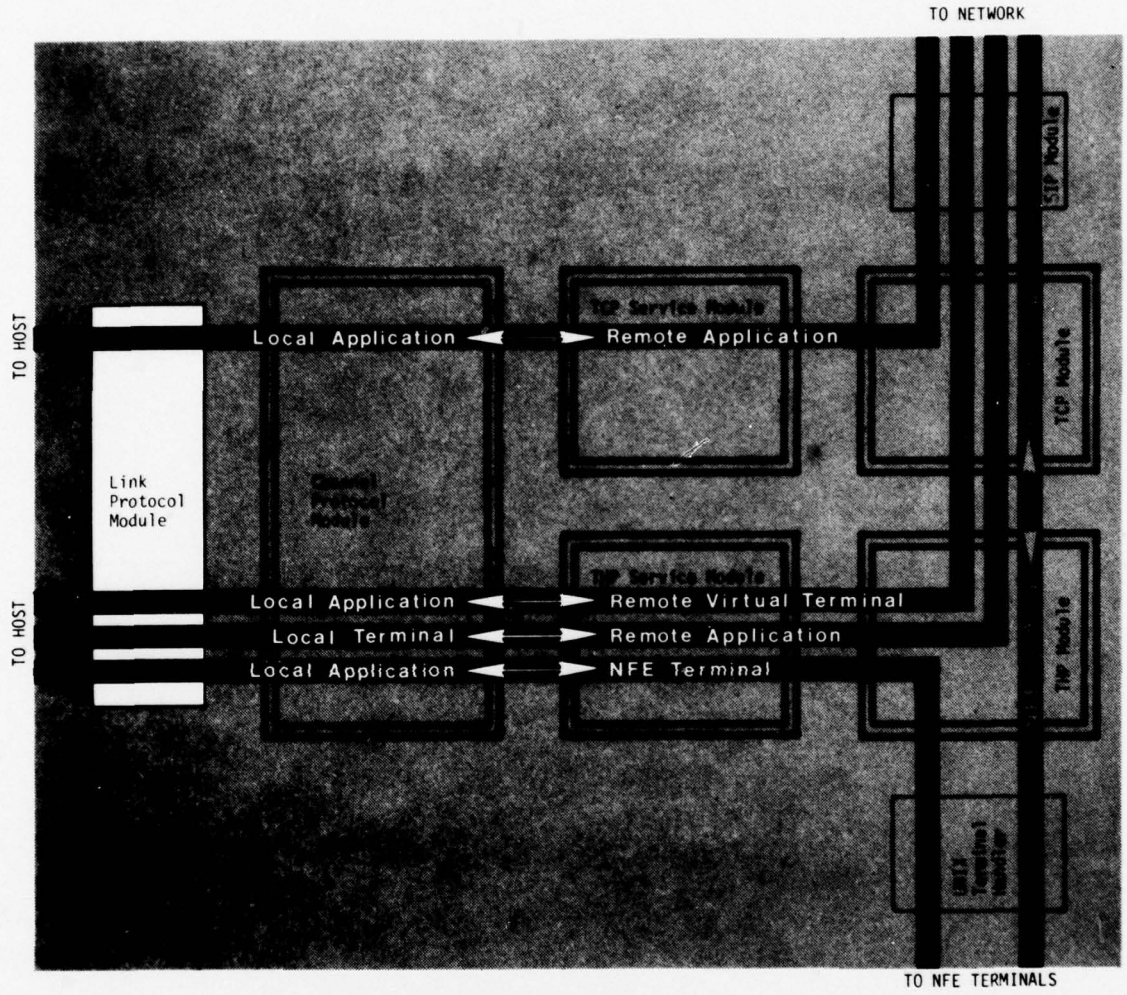


Figure 4: Place of the Link Protocol Module

Link Protocol Module

LEVEL:

UNIX kernel

TYPE:

Memory Segment Device Driver

IMPLEMENTS:

Link Level Host-Frontend Protocol

FUNCTION:

Establishing and maintaining communication between the Channel Protocol Modules in the local host and in the frontend.

ROUTES:

Local Application - Remote Application
Local Application - Remote Virtual Terminal
Local Virtual Terminal - Remote Application
Local Application - NFE Terminal

COMMUNICATES WITH:	VIA:
local host	host-frontend hardware interface
Channel Protocol Module	memory segment I/O interface

REMARKS:

The Link Level Host-Frontend Protocol was implemented in the ENFE by the H6000 to PDP-11 Computer Link Conventions (Alsberg et al. 1977a:17-19). This implementation will be carried over into the INFE essentially unchanged.

The place of the Link Protocol Module in the INFE software architecture is shown in Figure 4, opposite.

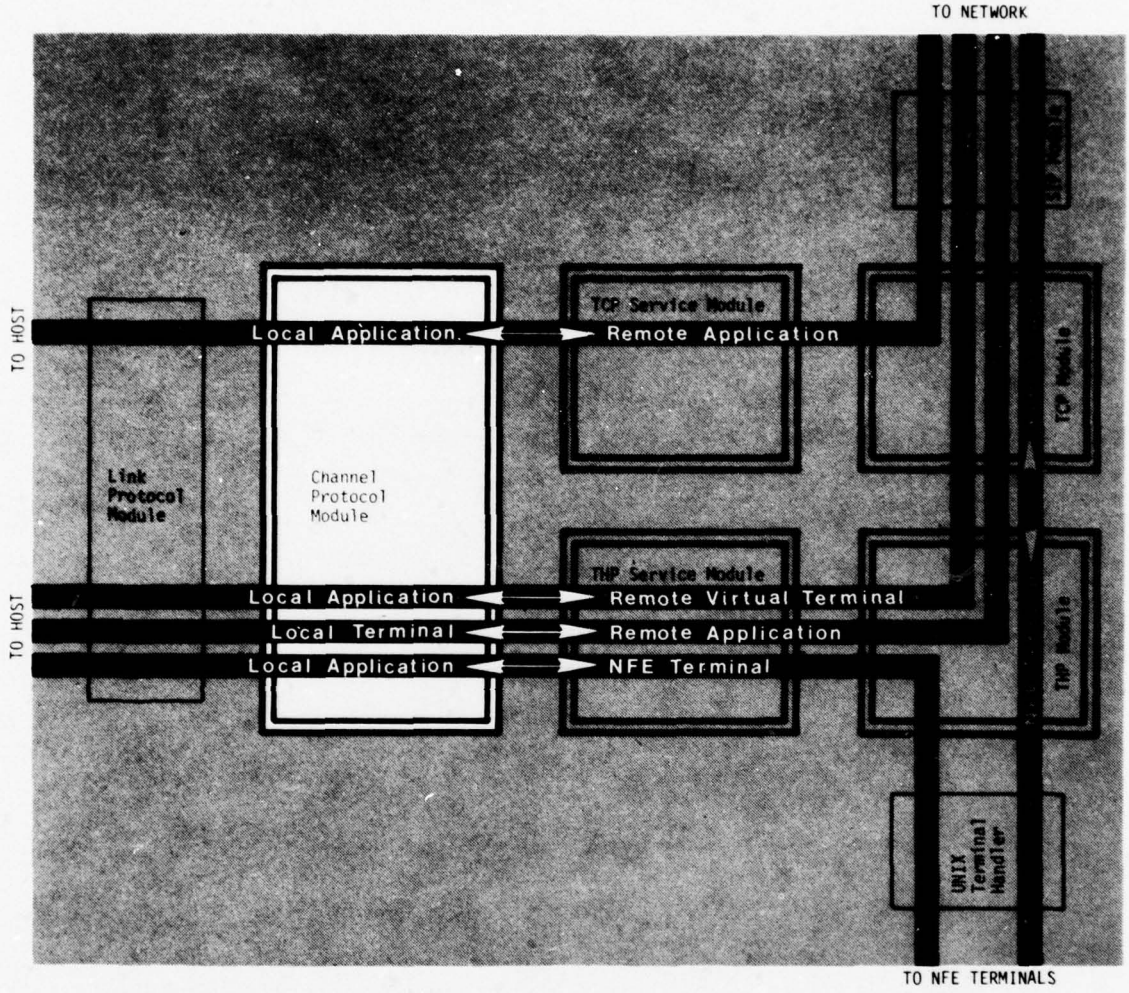


Figure 5: Place of the Channel Protocol Module

Channel Protocol Module

LEVEL: UNIX user

TYPE: Process

IMPLEMENTS:

Channel Level Host-Frontend Protocol

FUNCTION:

Initializing, maintaining, and terminating logical channels between host and frontend processes.

ROUTES:

Local Application - Remote Application
Local Application - Remote Virtual Terminal
Local Virtual Terminal - Remote Application
Local Application - NFE Terminal

COMMUNICATES WITH:	VIA:
Link Protocol Module	memory segment I/O
TCP Service Module	IIPC interface
THP Service Module	IIPC interface

REMARKS:

The Channel Level Host-Frontend Protocol was implemented in the ENFE by the Channel Protocol Module (Holmgren et al. 1977a:20-46). This implementation will be carried over into the INFE essentially unchanged.

The place of the Channel Protocol Module in the INFE software architecture is shown in Figure 5, opposite.

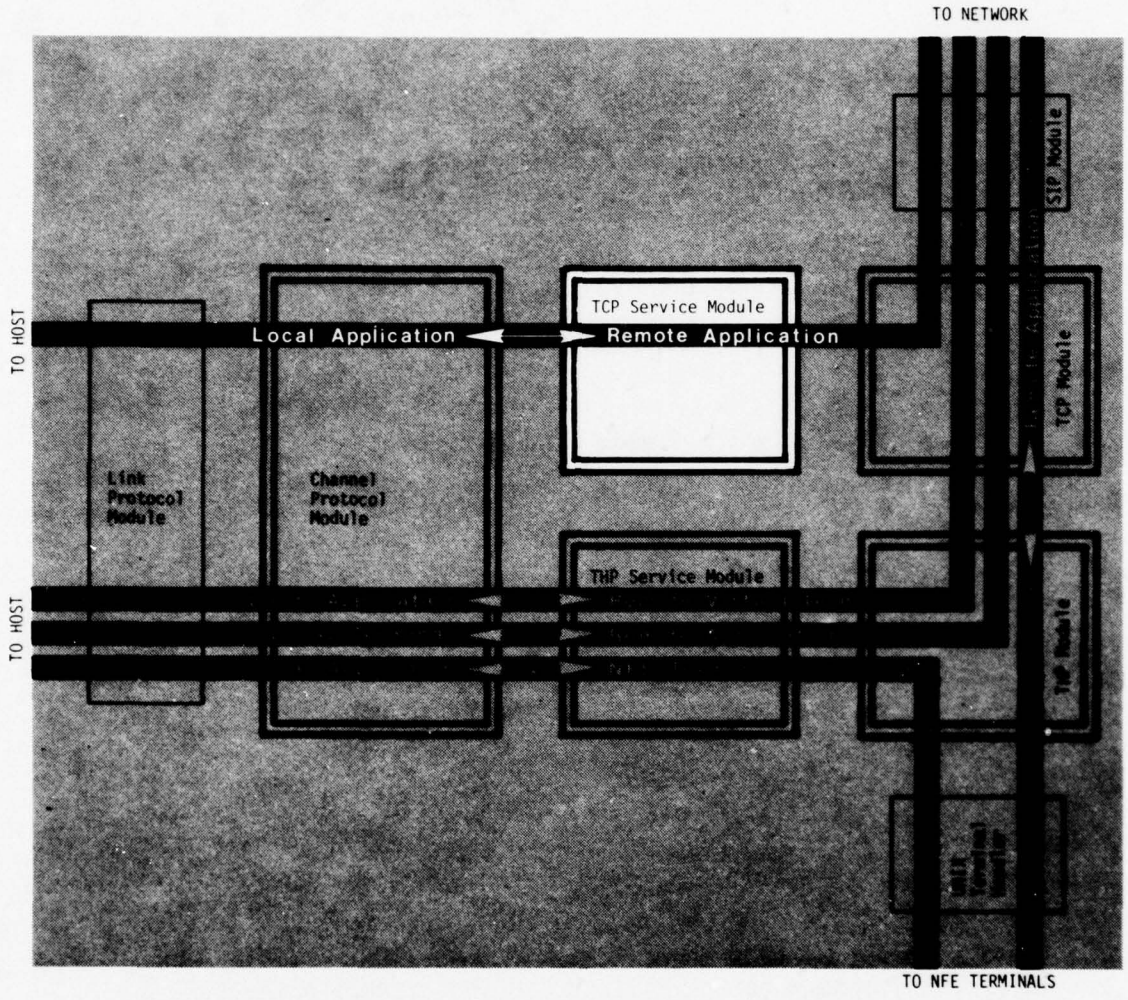


Figure 6: Place of the TCP Service Module

TCP Service Module

LEVEL:

UNIX user

TYPE:

Process

IMPLEMENTS:

TCP Process-Service Level Host-Frontend Protocol

FUNCTION:

Translating between HFP and TCP formats.

ROUTE:

Local Application - Remote Application

COMMUNICATES WITH:	VIA:
Channel Protocol Module	IIPC interface
TCP Module	IIPC I/O interface

REMARKS:

The TCP Service Module replaces the ENFE Host-to-Host Service Module (Holmgren et al. 1977a:77-88).

The place of the TCP Service Module in the INFE software architecture is shown in Figure 6, opposite.

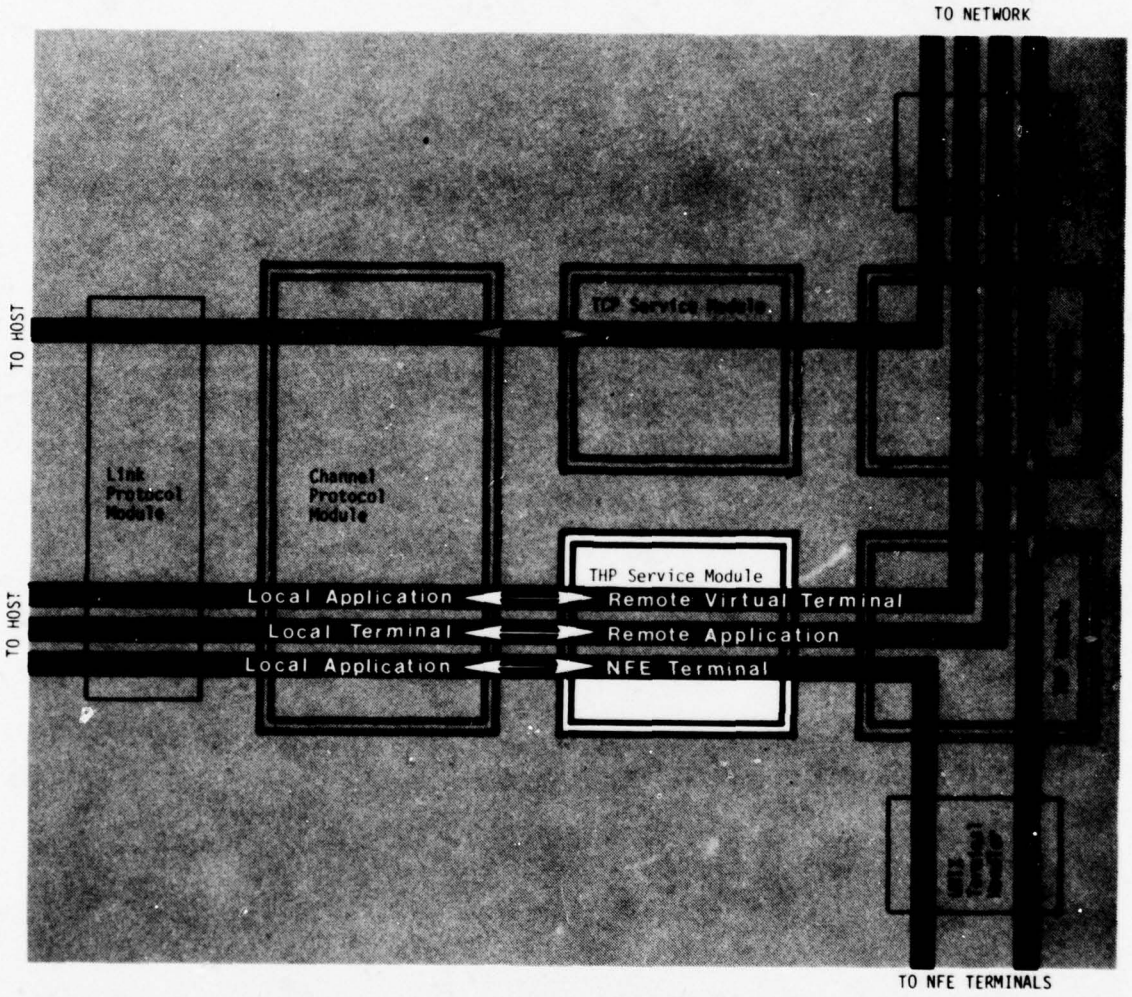


Figure 7: Place of the THP Service Module

THP Service Module

LEVEL:

UNIX user

TYPE:

Process

IMPLEMENTS:

THP Process-Service Level Host-Frontend Protocol

FUNCTION:

Translating between HFP and THP formats.

ROUTES:

Local Application - Remote Virtual Terminal
Local Virtual Terminal - Remote Application
Local Application - NFE Terminal

COMMUNICATES WITH:	VIA:
Channel Protocol Module	IIPC interface
THP Module	IIPC I/O interface

REMARKS:

The THP Service Module replaces the ENFE Program Access Service Module (Holmgren et al. 1977a:118-132) and the Channel Protocol Module interface part of the Server Virtual Terminal Service Module (Holmgren et al. 1977a:90-102).

The place of the THP Service Module in the INFE software architecture is shown in Figure 7, opposite.

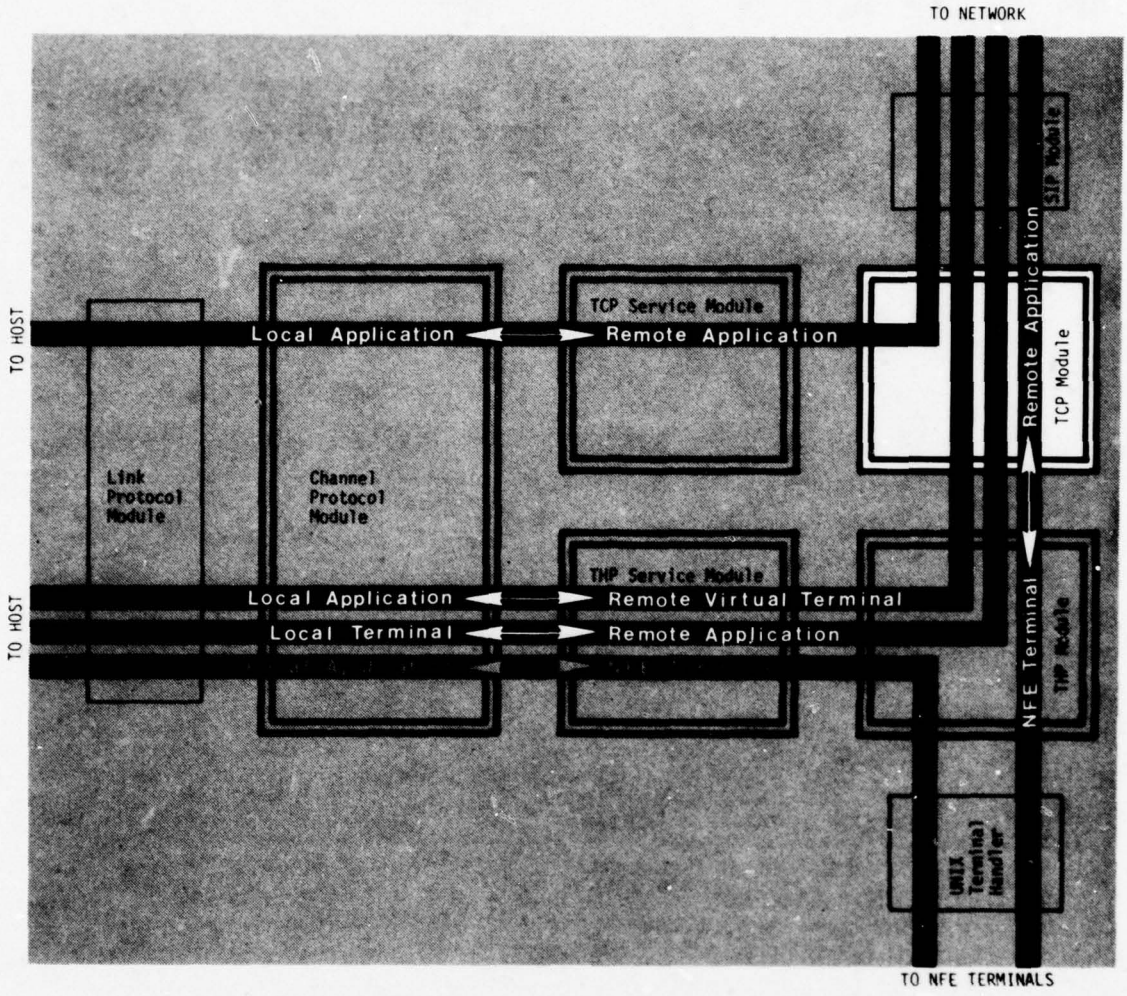


Figure 8: Place of the TCP Module

TCP Module

LEVEL: UNIX user

TYPE: Process

IMPLEMENTS:

Transmission Control Protocol

FUNCTION:

Transferring data reliably via the network.

ROUTES:

Local Application - Remote Application
Local Application - Remote Virtual Terminal
Local Virtual Terminal - Remote Application
NFE Terminal - Remote Application

COMMUNICATES WITH:	VIA:
TCP Service Module	IIPC I/O interface
THP Module	IIPC I/O interface
SIP Module	memory segment I/O interface

REMARKS:

The TCP Module replaces the ENFE NCP Daemon (undocumented) and the NCP Software (undocumented).

The place of the TCP Module in the INFE software architecture is shown in Figure 8, opposite.

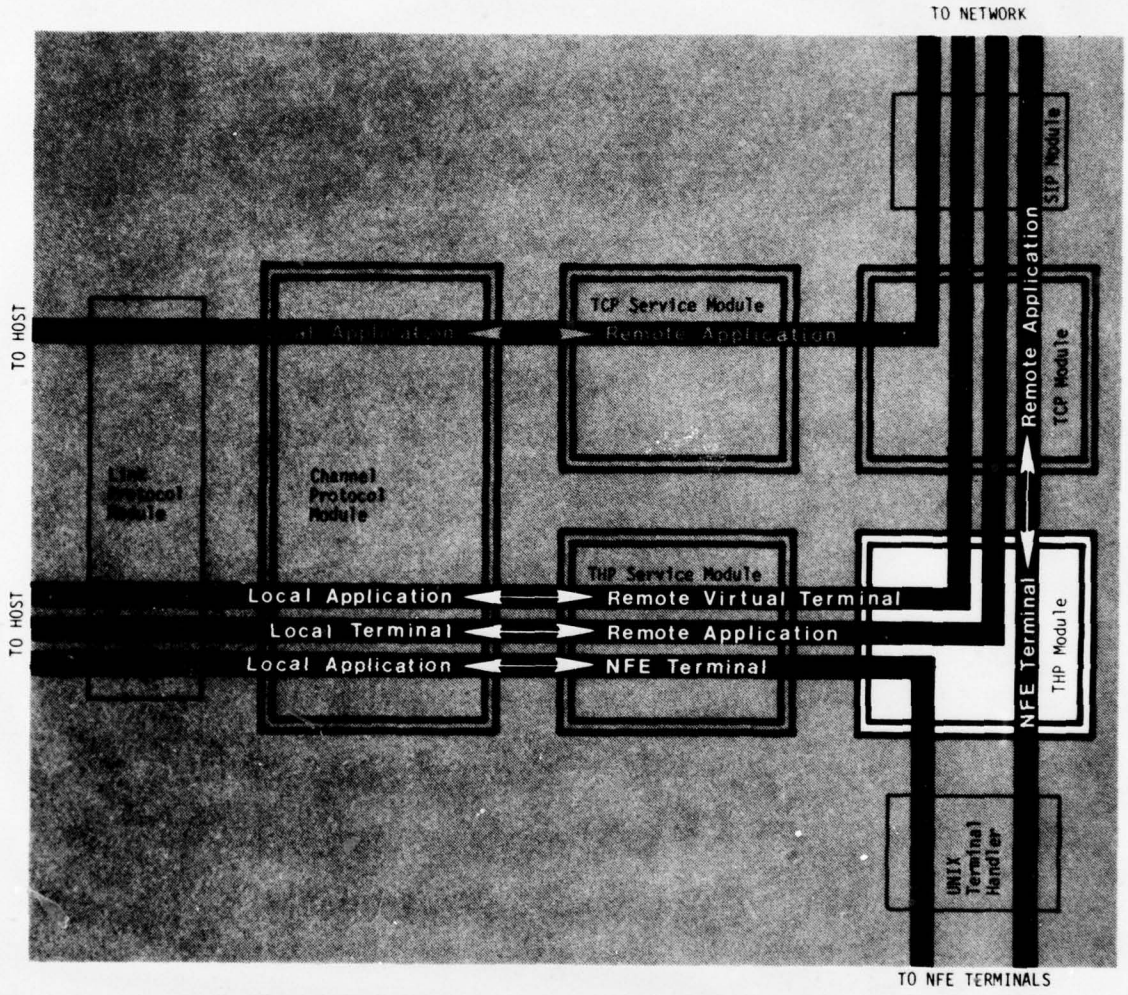


Figure 9: Place of the THP Module

THP Module

LEVEL:

UNIX user

TYPE:

Process

IMPLEMENTS:

Terminal-Host Protocol

FUNCTION:

Translating between NFE terminal representations and the network virtual terminal representation.

ROUTES:

Local Application - Remote Virtual Terminal

Local Virtual Terminal - Remote Application

Local Application - NFE Terminal

NFE Terminal - Remote Application

COMMUNICATES WITH:	VIA:
THP Service Module	IIPC I/O interface
Unix Terminal Handler	terminal non-blocking I/O interface
TCP Module	IIPC I/O interface

REMARKS:

The THP Module is to the INFE for AUTODIN II what the User Telnet Module (undocumented) and the Telnet Handler part and NCP interface part of the Server Virtual Terminal Service Module (Holmgren 1977a:90-116) were to the ENFE for the ARPANET.

The place of the THP Module in the INFE software architecture is shown in Figure 9, opposite.

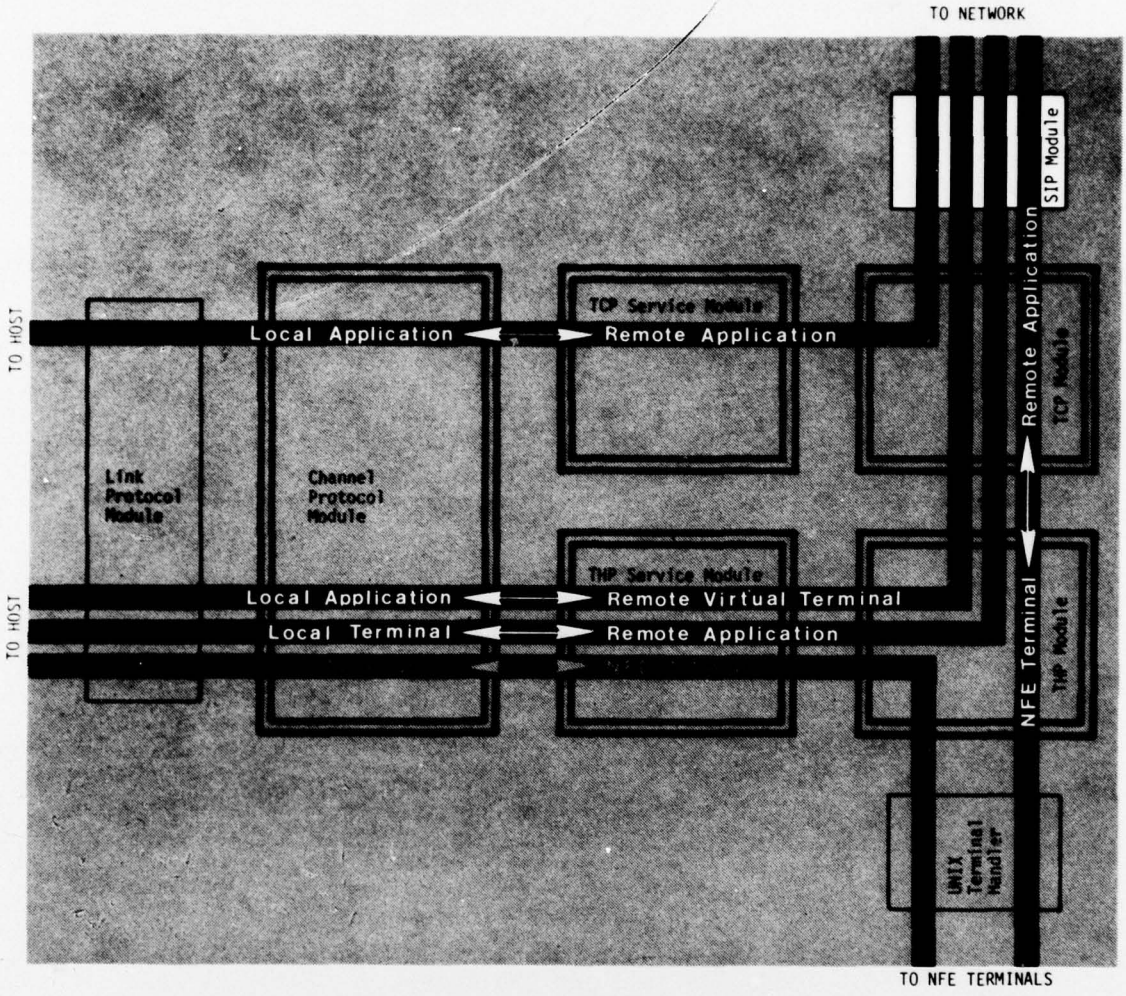


Figure 10: Place of the SIP Module

SIP Module

LEVEL:

UNIX kernel

TYPE:

Memory Segment Device Driver

IMPLEMENTS:

Segment Interface Protocol

FUNCTION:

Establishing and maintaining communication with the local packet switch.

ROUTES:

Local Application - Remote Application
Local Application - Remote Virtual Terminal
Local Virtual Terminal - Remote Application
NFE Terminal - Remote Application

COMMUNICATES WITH:	VIA:
TCP Module	memory segment I/O interface
local packet switch	network-frontend hardware interface

REMARKS:

Pending AUTODIN II availability, a Pseudo Segment Interface Protocol will be implemented using the ARPANET IMP-Host Protocol.

The place of the SIP Module in the INFE software architecture is shown in Figure 10, opposite.

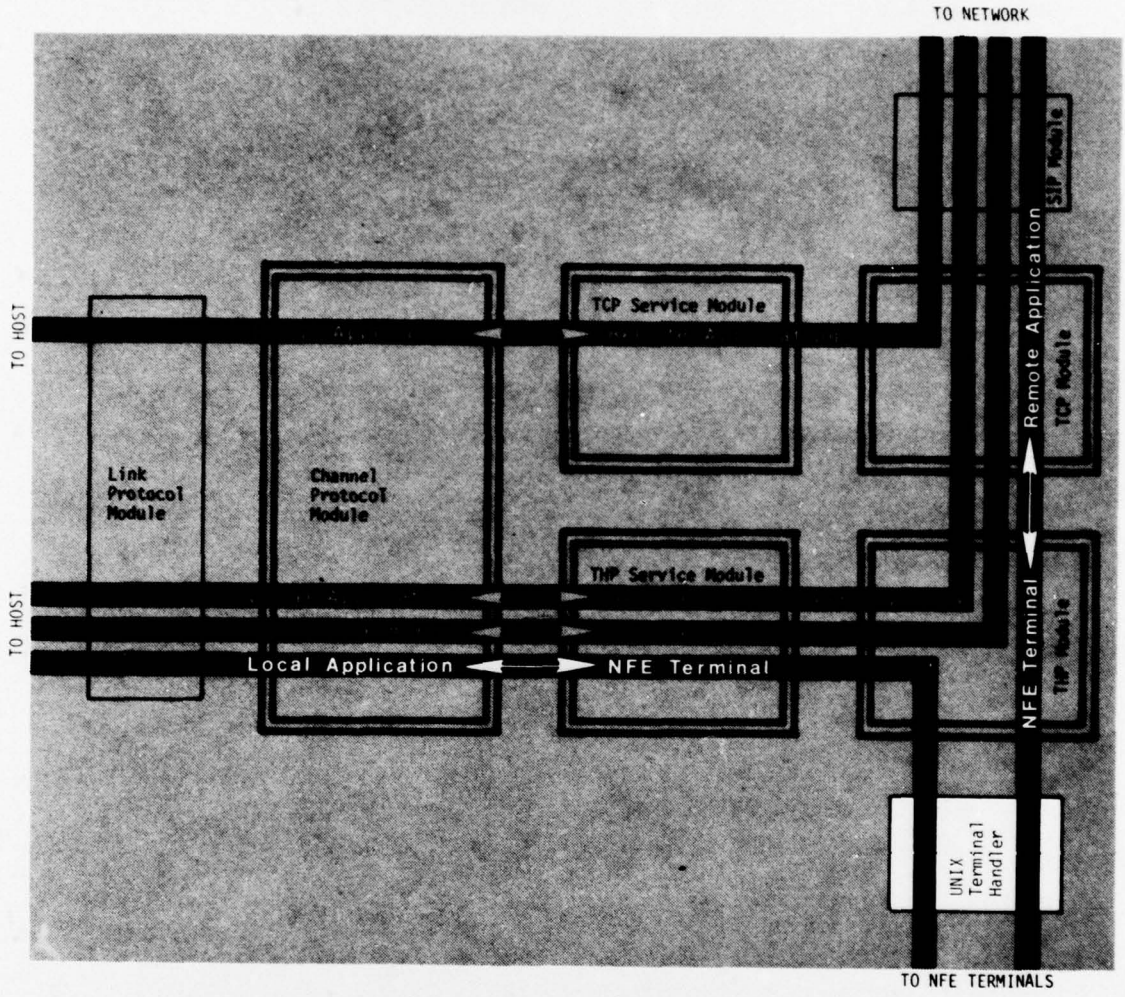


Figure 11: Place of the UNIX Terminal Handler

UNIX Terminal Handler

LEVEL:

UNIX kernel

TYPE:

Character Device Driver

IMPLEMENTS:

UNIX Terminal Protocol

FUNCTION:

Translating between the UNIX terminal representation and the representations of terminals attached to the INFE.

ROUTES:

Local Application - NFE Terminal

NFE Terminal - Remote Application

COMMUNICATES WITH:	VIA:
THP Module	terminal non-blocking I/O interface
Terminals	terminal-frontend hardware interface

REMARKS:

The UNIX Terminal Handler for the ENFE will be carried over into the INFE essentially unchanged.

The place of the UNIX Terminal Handler in the INFE software architecture is shown in Figure 11, opposite.

INFE DATA FLOW:

Protocol Staging along the INFE Routes

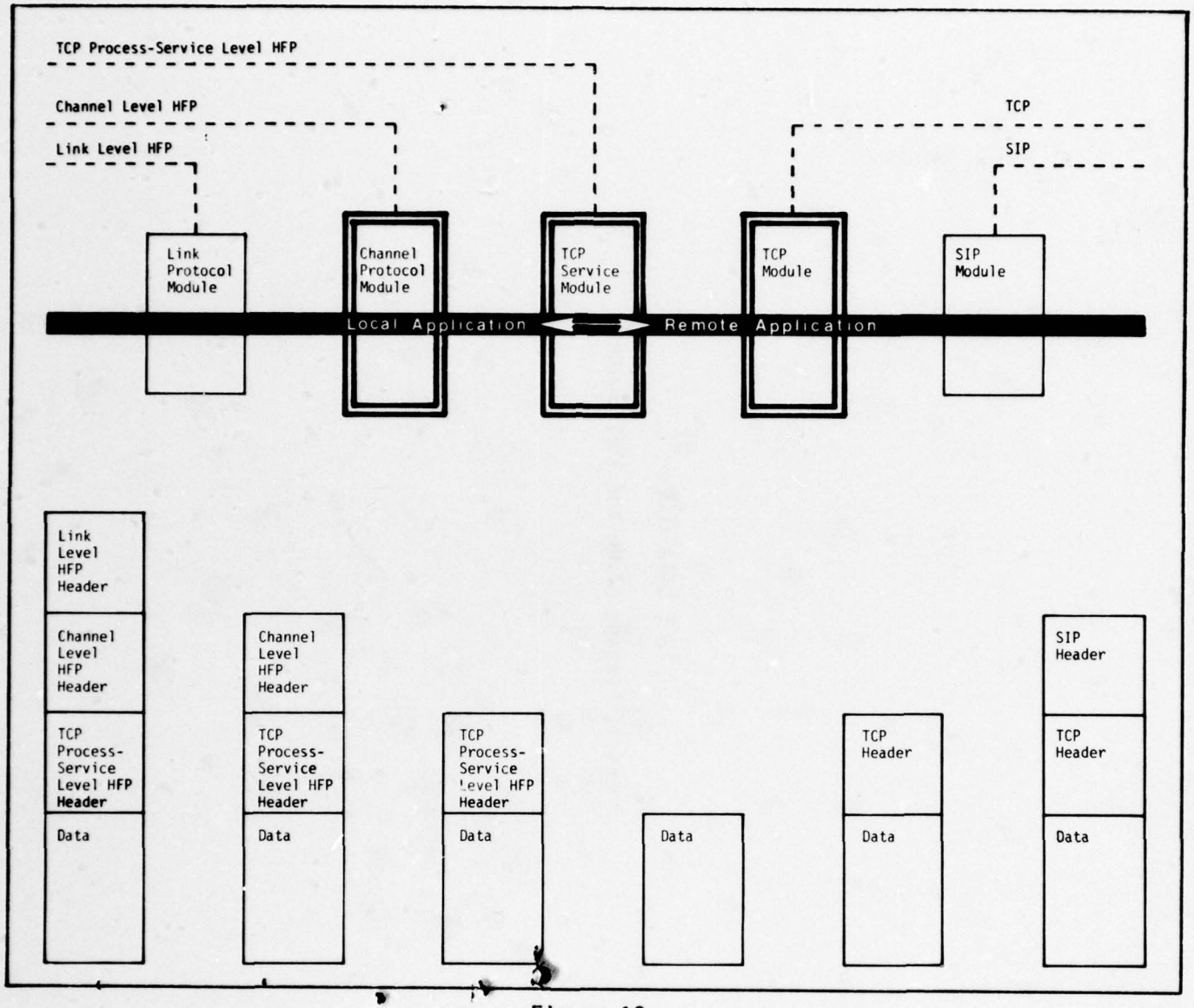


Figure 12:

Protocol Staging along the Local Application - Remote Application Route

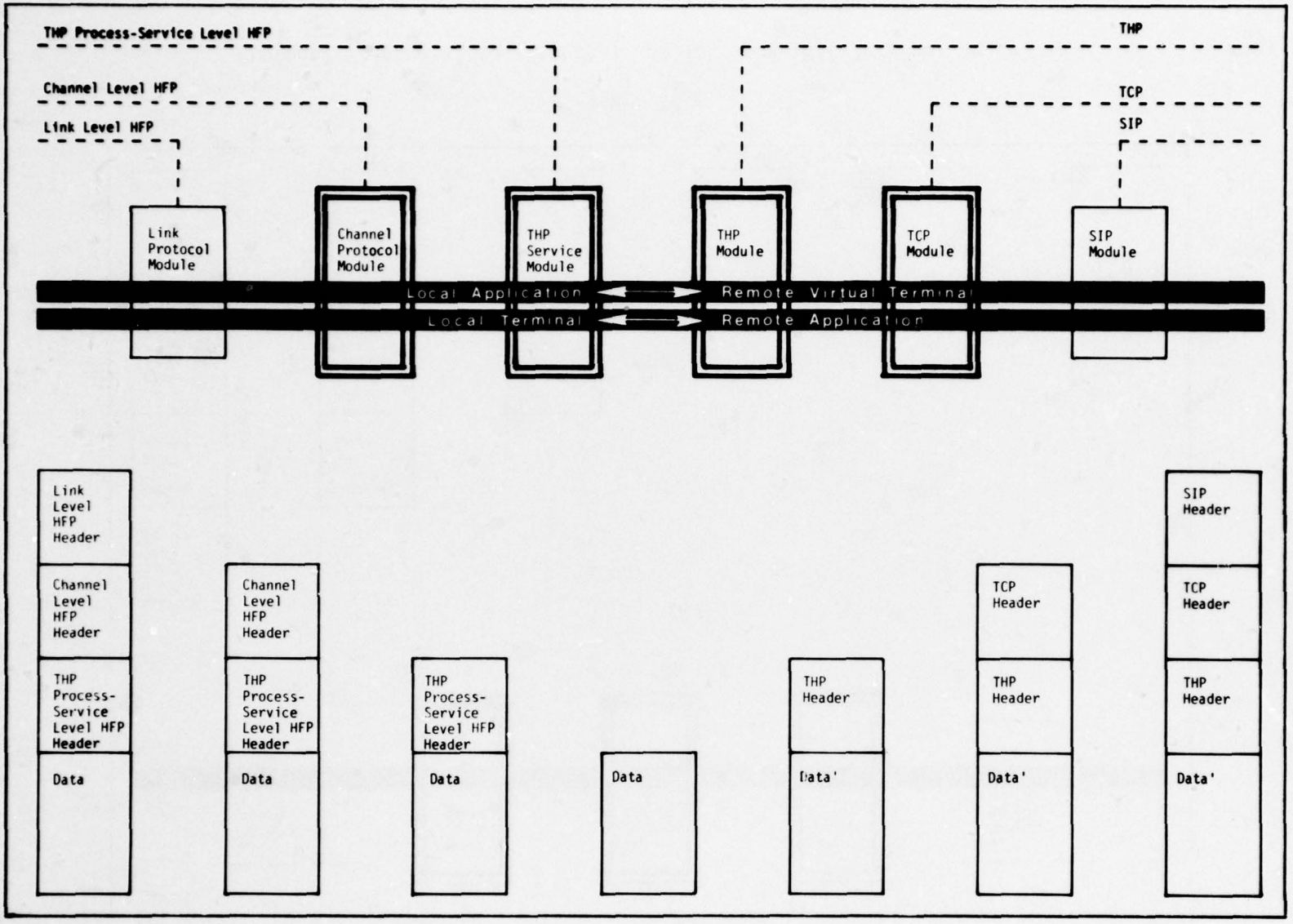


Figure 13:

Protocol Staging along the Local Application - Remote Virtual Terminal and the Local Terminal - Remote Application Routes

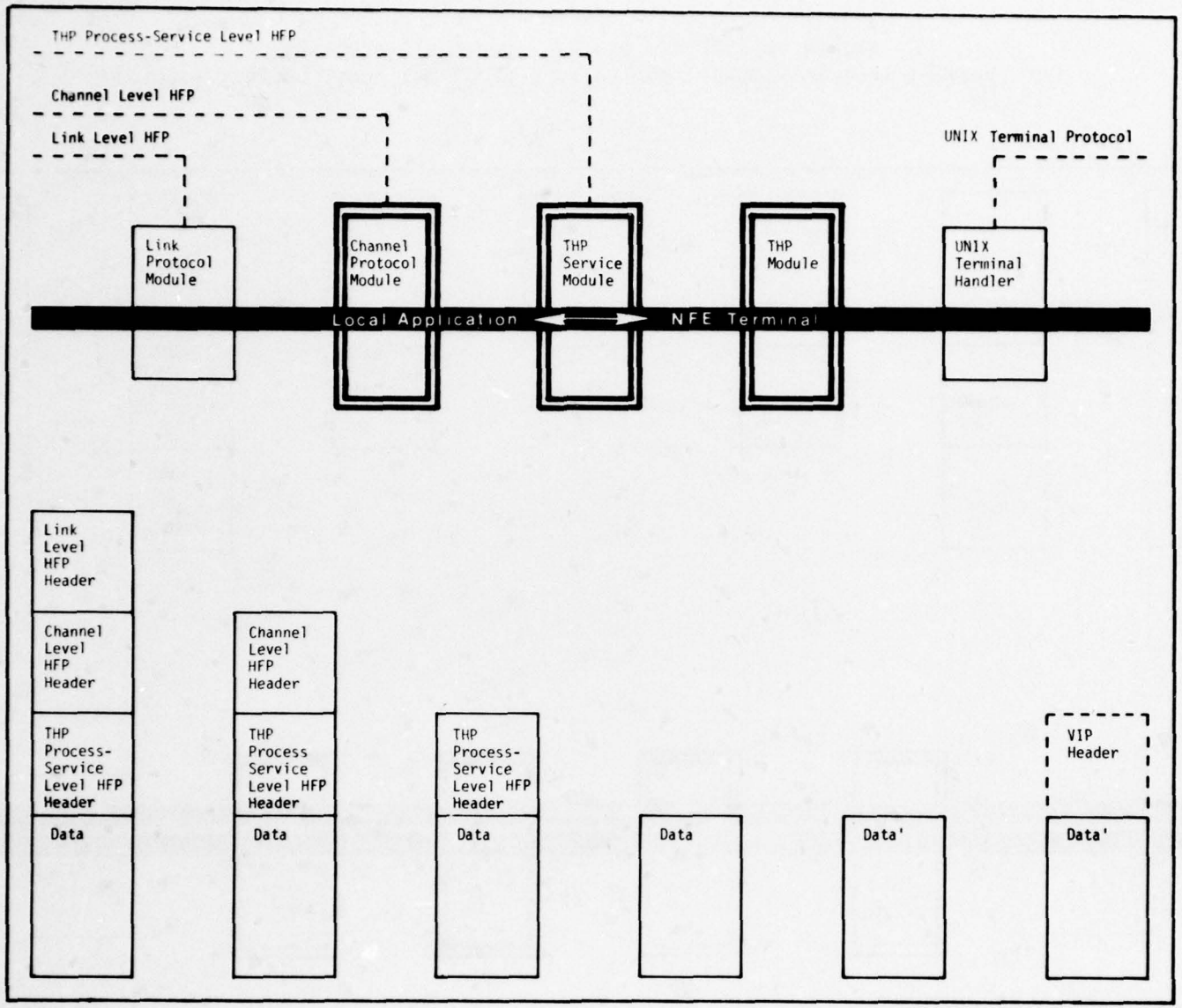


Figure 14:

Protocol Staging along the Local Application - NFE Terminal Route

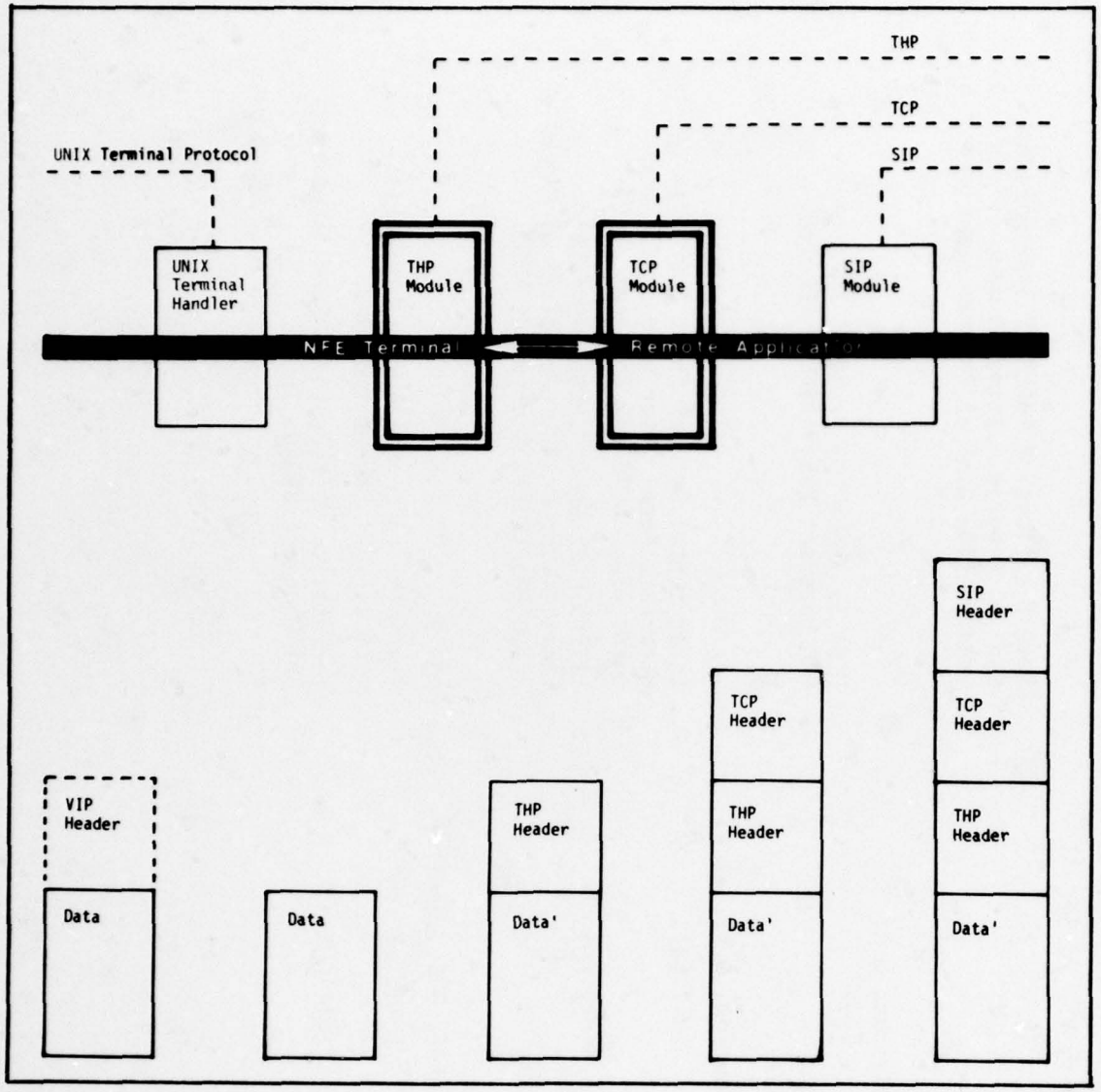


Figure 15:

Protocol Staging along the NFE Terminal - Remote Application Route

REFERENCES

Alsberg, P.A., et al.

1977 — H6000 Software Specifications. CAC Document 220; CCTC-WAD Document 7501.

Cerf, V., Postel, J.

1977 — Specifications of INTERNET Transmission Control Program. Internal memo of the Transmission Control Program committee.

Grossman, G.R.

1976 — Host to Front End Protocol (draft). CAC Document 219; ARPANET RFC 710.

Holmgren, S.F., et al.

1977 (a) — Experimental Network Front End Software Functional Description. CAC Document 233; CCTC-WAD Document 7515.

1977 (b) — Illinois Inter-Process Communication Facility for UNIX. CAC Technical Memorandum 84; CCTC-WAD Document 7507.

Postel, J.B., Garlick, L.L., Rom, R.

1976 (a) — Transmission Control Protocol Specification. SRI Document 35938.

1976 (b) — Terminal to Host Protocol Specification. SRI Document 35940.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTI Document Number 2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* INFE Software Functional Description Overview		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER DTI Document Number 2
7. AUTHOR(s) Gary R. Grossman Steven F. Holmgren Richard H. Howe		8. CONTRACT OR GRANT NUMBER(s) DCA100-77-C-0069
9. PERFORMING ORGANIZATION NAME AND ADDRESS Digital Technology Incorporated 302 East John, Champaign IL 61820		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DCA/Command and Control Technical Center ATTN C420 11440 Isaac Newton Square, North Reston, VA 22090		12. REPORT DATE March 20, 1978 13. NUMBER OF PAGES 30
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release;
distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

No restriction distribution.

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Network Frontend

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Digital Technology Incorporated is developing an IOC Network Frontend (INFE) to connect a WWMCCS H6000 to AUTODIN II. A DEC PDP-11/70 running under a modified Network UNIX system is being used for the INFE. This document describes the data routes through the INFE, the INFE protocol modules, and the flow of data vis-a-vis protocol staging in the INFE.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*